ARMY RESEARCH LABORATORY

Juxtaposition of Inertial Navigation Sensor and Camera Egomotion Estimates of Ground Vehicle Trajectory: Results and Implementation Details

by Gary A. Haas and William F. Oberle

ARL-TR-3096 **October 2003**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-3096 **October 2003**

Juxtaposition of Inertial Navigation Sensor and Camera Egomotion Estimates of Ground Vehicle Trajectory: Results and Implementation Details

Gary A. Haas and William F. Oberle
Directorate, ARL

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
October 2003	Final	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Juxtaposition of Inertial Navigation Sensor and Camera Egomotion Estimates of Ground Vehicle Trajectory: Results and Implementation Details	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Haas, G.A.; Oberle, W.F. (both of ARL)	AH80
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory Weapons and Materials Research Directorate Aberdeen Proving Ground, MD 21005-5066	ARL-TR-3096

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Accurate knowledge of the pose (position and orientation) is essential to the unmanned ground vehicle's (UGV) ability to follow a prescribed path and to reach a specified location, which is the essence of autonomous mobility. An essential element of an autonomous UGV's sensor suite is the navigation sensor. Given appropriate processing, other UGV terrain sensors can also provide independent estimates of vehicle pose or derivatives thereof that can be fused with the navigation data to provide a more robust knowledge of pose and implicitly, of registration between the two sensors. Improved accuracy of UGV pose in turn enables improved accuracy of terrain geometry estimates.

In this report, we describe a precursor to fusion, which is juxtaposition of consistent entities ("apples to apples"). Trajectories of pose estimates are recorded as a surrogate UGV traverses rough terrain, one trajectory from an inertial UGV navigation sensor, and its counterpart from egomotion extracted in post-processing from a forward looking video camera. The sensors and the relevant aspects of the data collection methodology are described, as are the streams of pose estimates. We explain the procedure adopted for temporally aligning the data streams and the assumptions concerning registration under which the streams are juxtaposed. A notation is developed to relate the myriad coordinate frames implicit in the application. The juxtaposition is presented graphically, with recommendations for subsequent studies with the intent of achieving fusion. A small related study evaluating the possibility of calculating the registration from pose trajectories is reported.

15. SUBJECT TERMS

Egomotion, navigation data (NAV), sensor fusion, sensor registration, UGV

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UL		Gary A. Haas
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (*Include area code*) 410-278-8867

Contents

List of Figures iv

List of Tables iv

1. Introduction 1

2. Purpose 1

3. The Data 2

4. The Sensors 3

4.1 Egomotion Extraction....4

5. Sensor Registration 6

6. Pre-processing 6

6.1 Temporal Alignment 7

6.2 Consistent Representation 7

6.2.1 Notation 8

6.2.2 Representation 8

6.2.3 Manipulation 9

6.3 Visualization....10

7. Results 10

8. Registration From Trajectory 13

8.1 Registration From Trajectory With Synthetic Data 14

9. Conclusions 16

10. References 17

Appendix A. MATLAB Code to Plot Juxtaposed Translations 18

Appendix B. MATLAB Code to Plot Juxtaposed Incremental Rotations 21

Appendix C. MATLAB Code to Plot Juxtaposed Absolute Rotations 23

List of Figures

Figure 1. Three frames of the handrail sequence. 2

Figure 2. Comparison of incremental rotations. (Roll measured by the nav system exceeded 10 degrees and pitch by 5 degrees during this sequence.) 11

Figure 3. XY plots of incremental rotations. 11

Figure 4. Comparison in world coordinates. 12

Figure 5. Comparison of incremental translation. 13

Figure 6. Registration of synthetic sensor from trajectory. 15

Figure 7. Registration of synthetic sensor from the same trajectory purged of motions near the known registration axis. (The magnitude of the computed registration rotation is approximately 1 degree, as expected, and the direction is almost pure pitch, again as expected.) 15

List of Tables

Table 1. Template for figures 2 and 4. 10

1. Introduction

An autonomous unmanned ground vehicle (UGV) finds its way through terrain to a destination point by means of on-board sensors. Common sensors include one or more of stereo video, imaging LAser Detection And Ranging (LADAR), and navigation. An important component of the UGV's world model is vehicle pose (position and rotation with respect to world coordinate axes) and the trajectory of the pose through time. Accurate knowledge of the pose is essential to the UGV's ability to follow a prescribed path and to reach a specified location—the essence of autonomous mobility. Knowledge of the pose is also necessary to integrate and correlate sensor information obtained at different times and from different poses as the UGV traverses the terrain. An essential element of the UGV's sensor suite is the navigation sensor, which is dedicated to measuring its pose. Given appropriate processing, other sensors can provide estimates of vehicle pose or derivatives thereof. This report describes the comparison of pose estimates from a UGV navigation sensor with the egomotion (motion of the sensor itself) extracted from imagery recorded by a forward looking video camera on board the vehicle.

2. Purpose

The study was undertaken to explore implementation issues in fusing and integrating multi-sensor data from a UGV. Juxtaposition is the first step in fusion and requires that a number of issues be taken into account so that fusion is between consistent entities, i.e., "apples to apples". Substantial effort has been devoted to collecting multi-sensor data in the UGV context (Shneier, 2003), but few UGV implementations have taken advantage of synergies made possible by multiple sensors. The initial focus of this study was to establish confidence in the rotational egomotion extracted from forward looking camera imagery by a new implementation created by the U.S. Army Research Laboratory (ARL). Confidence was to have been established by a comparison of the egomotion estimates to the rotational motion sensed by a navigation sensor (described in section 4) that was believed to be reasonably accurate. If the estimates of rotation were in substantial agreement, then the intent was to use measurements of translation from the navigation system to instantiate egomotion estimates of translation.

3. The Data

The sensor data described in this report are extracted from a larger data set collected over two days in May 2003 at the ARL test facility at Fort Indiantown Gap (FITG), Pennsylvania, in support of ARL's Collaborative Technology Alliance for Robotics. More than 100 gigabytes of data were collected from a manned HMMWV[1] (high mobility multi-purpose wheeled vehicle) equipped with a UGV sensor suite similar to that of the XUV (experimental unmanned vehicle) of ARL's UGV Demo III (Shoemaker and Bornstein, 1998). This HMMWV is managed and operated by personnel of the National Institute of Standards and Technology and is configured specifically for the collection of sensor data. From this collection, 6.7 seconds of data (200 video frames long) were selected for this comparison and were dubbed the "handrail sequence," or HS. The criteria used to select this sequence were

1. The imagery was to be representative of the data set, that is largely unstructured, dominated by unmarked gravel roads and vegetation. The HS included gravel, bushes, trees, and grass.
2. The subset was to include some structure visible through most of the trajectory to enable identification of consistent features from frame to frame. The HS featured a small wooden railing visible throughout the entire trajectory.
3. The subset must include variation in yaw, pitch, and roll so that measurements of all rotational axes can be evaluated. The HS was collected as the vehicle traversed a sweeping turn over rutted and uneven terrain, which satisfied the criterion.

Several frames spanning this subset are shown in figure 1.

Figure 1. Three frames of the handrail sequence.

[1] A military truck.

The sensor data were collected and stored by several computers that are coordinated through an on-board ethernet network. Data are not processed on line but are saved to disk in association with the precise time of acquisition (a process known as "time tagging"), to be analyzed off line after the collection. The clocks of the various computers are roughly synchronized before a data collection via the network time protocol, but the exact point in the recording process identified by the time tag is not well defined in all cases. For practical purposes, synchronization of the data is controlled by a common command to initiate collection of a data sequence (3 minutes in length, for the May 2003 FITG data collection), which is carried by the network to each of the data collection computers. The accuracy of the synchronization is affected by lags in the network and by other unevaluated lags in the recording and sensing of phenomena by the various sensors. Synchronization through time tags is at best a rough one, despite the millisecond resolution, and is discussed further in section 6.1.

4. The Sensors

The vehicle was equipped with an imaging LADAR unit, video cameras in a binocular stereoscopic configuration, a navigation (nav) unit based on the modular azimuth pointing system (MAPS) adapted from a self-propelled howitzer, a GPS (global positioning system) receiver, and several low resolution video cameras. The LADAR, GPS, and low resolution cameras were not used in the study and are not further described. The video cameras were used in this study as stand-alone vision sensors, not as a stereo pair.

The navigation unit is built around a Honeywell MAPS, which is an inertial measurement unit. The MAPS contains precision ring laser "gyros"[2], enabling the unit to track the orientation of the unit with respect to true north and gravity ("world coordinates") at a claimed accuracy of 0.5 milliradian (Honeywell, no date). Position information from the unit is calculated internally on the basis of dead reckoning[3]. GPS-augmented dead reckoning is available in the data set, but simple dead reckoning data were deemed more suitable for the study since the GPS-augmented data are prone to occasional "jumps". The dead reckoning data are quite noisy, however, as are the odometry data upon which they are based. The most useful representation of the MAPS data relates the instantaneous location and orientation of the unit to global north, east, and down, available at 20 Hz. So-called "strap-down" data (instantaneous velocities and rotational velocities in sensor coordinates) were not available.

[2] A ring laser gyro is similar in function to a gyroscope, and the etymological roots of the term are apparent, but it is not phenomenologically a gyroscope. A gyroscope is a mechanical device whereas the ring laser gyro and its phenomenological cousin, the fiber optical gyro, are essentially optical devices.

[3] "Dead reckoning is the process of estimating your position by advancing a known position using course, speed, time, and distance to be traveled." (http://www.deadreckoning.com/dead_html/).

The cameras are three-chip 640- by 480-pixel resolution, CCD (charge-coupled device) technology, which use a progressive scan-like mode in which odd and even fields are "grabbed" simultaneously at 30 Hz. The output is analog RGB (red-green-blue) format and is digitized by a computer-mounted frame grabber board at 8 bits per pixel for each of red, green, and blue. The resulting data stream is stored in uncompressed digital form at full temporal and spatial resolution in a redundant array of inexpensive disks (RAID). The 8-mm lenses feature manual iris and focus so that a simple mathematical camera model can be used when warranted (e.g., for binocular stereo). Camera electronic iris and automatic gain control functions are enabled so that the cameras deliver quality imagery under a range of illumination. Frames are time tagged when digitized by the frame grabber, but the time tags exhibit substantial noise in comparison to the 30 Hz at which the frames are known to be acquired by the cameras. The actual timing of frame acquisition of both cameras is controlled by the internal timing circuitry of one of the cameras, conveyed to the other through the analog "genlock"[4] interface.

4.1 Egomotion Extraction

Each image frame of the video imagery is processed to extract egomotion from frame to frame. The egomotion for image frame *n* is represented as a spatial transformation of the camera coordinate frame[5] from the time when image *n* was acquired to the time image *n*+1 was acquired. The transformation is expressed as a rotation and translation of unit length in the coordinates of the camera at the time of image frame *n*. The translation is expressed as a unit vector because the true magnitude cannot be determined in the absence of a known distance in the scene depicted by the image (Trucco and Verri, 1998). The approach for computing the egomotion is based on an algorithm and software that we previously developed for stereo camera re-calibration (Oberle, 2003). Essentially, a four-step process is used. In the first step, a Harris corner detector (Harris and Stephens, 1988) is initially used to identify point sets in image frame *n* and *n*+1, which correspond to features associated with corners. Matching points in image frame *n* and *n*+1 are then identified with the following procedure. Each corner point identified in image *n* is compared to every corner point identified in image *n*+1. The comparison uses 3 by 3 correlation windows centered at the points with the metric the sum of squared differences (SSD). The corner point in image *n* is matched with the corner point in image *n*+1 that minimizes the value of the SSD. This produces an initial set of matched points where each element in the set consists of a point in image *n*, a matching point in image *n*+1, and the SSD value used to select the match. We construct the final set of matched points from this set by deleting half of the elements with the highest SSD values. The second step involves estimating the so-called "fundamental matrix" relating the two image frames. By definition (Trucco and Verri, 1998),

[4] A common camera interface enabling the synchronization of video signals from one or more cameras.

[5] Camera frame *n* associated with image *n* is a right-handed Cartesian coordinates system with the origin at the center or focus of projection using the perspective or pinhole camera model (Trucco and Verri, 1998, page 27). Unless otherwise stated, the term "frame" refers to this camera-based coordinate frame rather than to a "frame" from an image sequence.

if $\overline{p}_n$ and $\overline{p}_{n+1}$ are a pair of corresponding homogeneous points in pixel coordinates in the *n* and *n*+1 images, respectively, the fundamental matrix, $\mathbf{F}$, satisfies the equation

$$\overline{p}_{n+1}^{T}\mathbf{F}\overline{p}_n = 0. \quad (1)$$

The superscript T represents the matrix transpose. Thus, the matching points identified in the first step can be used to estimate the entries of **F**. Unfortunately, this calculation is extremely susceptible to noise in the location of the matching points, and straightforward approaches (e.g., matrix inversion) to solve the resulting system of equations generally yield poor results. To mitigate the impact of noisy input data in the calculation, a variation of the RANdom SAmple Consensus (RANSAC) paradigm suggested by Torr (2002), the Maximum *A Posteriori* SAmple Consensus (MAPSAC), followed by a constrained nonlinear estimator is used in the fundamental matrix calculation. The RANSAC paradigm attempts to minimize the impact of noisy data by randomly selecting a minimum sized subset (e.g., two points for a line, seven matched points for the fundamental matrix[6]) from the input data required to compute the fundamental matrix. This process is repeated a sufficient number of times to ensure that the probability that at least one subset will be comprised of only "good" data points (i.e., those that satisfy a given condition such as equation 1) exceeds a pre-set level, usually 95% or 98%. The best solution, i.e., estimated fundamental matrix, is the one that maximizes the number of data points satisfying the given condition. Instead of simply determining the number of points satisfying a given condition, the MAPSAC paradigm introduces a negative logarithmic likelihood function and selects as the best solution the fundamental matrix that minimizes this likelihood function; see Torr (2002) for details. The fundamental matrix from the MAPSAC process is based on seven data points and not the entire set of good data points. To obtain an estimate based on all the good data points, the MAPSAC calculation is followed by a constrained nonlinear estimation of the fundamental matrix. In our implementation, only the good data points associated with the MAPSAC-computed fundamental matrix are used. The constrained nonlinear estimation is an iterative approach in which the entries of the fundamental matrix are iteratively adjusted to minimize the sum of weighted residuals under the constraints that the determinant of $\mathbf{F}$ equals zero and the Frobenius norm of the top 2 by 2 sub-matrix of $\mathbf{F}$ equals 1. The fundamental matrix from the MAPSAC calculation is used as the initial estimate of $\mathbf{F}$. See Torr (chapter 4, 2002) for specific details concerning the weighted residuals. Step 3 in the egomotion calculation involves the estimation of the "essential" matrix from the fundamental matrix via the relationship (Trucco and Verri, 1998)

$$\mathbf{E} = \mathbf{M}^{T}\mathbf{F}\mathbf{M}. \quad (2)$$

The matrix $\mathbf{M}$ is the matrix of the "intrinsic" parameters (Trucco and Verri, 1998) for the camera, which are known quantities from the camera calibration. The essential matrix from

[6]Although the fundamental matrix has nine entries, its determinant must equal zero and one of its entries can be set to an arbitrary constant. Thus, there are only seven independent entries.

equation 2 is adjusted by means of singular value decomposition to ensure that the two essential matrix constraints (i.e., rank = 2 and nonzero singular values equal) are satisfied (Trucco and Verri, 1998). Finally, in step 4, the essential matrix from step 3 is factored by means of singular value decomposition (Wang and Tsui, 2000) to identify candidate egomotion rotations and translations of unit length. Back projection of a single image point in image *n* is used to determine the appropriate rotation-translation combination consistent with the input image information.

5. Sensor Registration

Each sensor has its own native coordinate frame, as does the vehicular platform to which the sensor subsystems are mounted. As the vehicle follows a trajectory through space, the data stream from each sensor describes the trajectory of the sensor in its own coordinate frame. The determination of the transformations among the coordinate frames of the various sensors is a process known as "registration". The transformations defined by the registration process are also known as "the registration". The sensors of this study were not registered in any meaningful fashion, although a rough registration of the stereo rig and the LADAR imager was described in Oberle and Haas (2002). It is known that the camera was pointed roughly "forward" in the direction of travel of the data collection vehicle and was approximately upright. It is also known that the navigation sensor is similarly aligned with the direction of travel of the vehicle and that it senses gravity directly. So there is an approximate alignment between the two coordinate frames, but there remains an unknown translation and (more important) rotation between them. One intriguing possibility is that the rotational component of the registration between the sensors can be extracted from the rotational difference between each sensor's representation of its trajectory. This is discussed in more detail in section 8.

6. Pre-processing

Several decisions had to be made in the preparation of the data from the two sensors to facilitate comparison. First, the data had to be aligned temporally, as described in section 6.1. Second, the data had to be placed in a consistent framework to allow comparison, as described in section 6.2. Finally, the data had to be depicted so that similarities and differences were apparent to the researcher, as described in section 6.3.

6.1 Temporal Alignment

When stored sensor data from different sensors are reassembled for purposes of multi-sensor fusion and integration, the time tags are used for temporal alignment relative to the initiation of recording a data sequence. The functional time base for a data element in a sequence is the difference in time between its time tag and the time tag value for the first element of the data sequence. As observed previously, several problems existed in the time base of the HS data set. The imagery time tags showed substantial jitter, the two data records were tagged from different clocks, and the internal sampling rates of the two sensors are different. For purposes of temporal alignment of a sequence of images with the nav sensor, the time tag was assumed to be correct for the first image of the sequence, and the rest of the images were assumed to occur at exact 30-Hz intervals thereafter. An unknown but roughly constant offset in the time base was thus substituted for the random frame-by-frame error in the imagery time tags. To obtain pose estimates at corresponding times, re-sampling of the nav data was performed with linear interpolation. Interpolation of translations is straightforward, and the interpolation of rotations is performed on the quaternion representation of the rotation, as suggested by Shoemake (1985). Inspection of initial results of the resulting data alignment strongly suggested that nav data lagged the egomotion data by as much as 100 milliseconds (ms), so a global offset was introduced in the nav time base. That is, the nav data corresponding to an egomotion estimate at time *t* were interpolated from the measured nav data points with time tags just before and just after time *(t + offset)*. The value of the global offset (30 ms) was selected because it yielded the smallest cumulative error (over the entire subset) in estimated rotational magnitude between the egomotion estimate and the corresponding interpolated nav value. The decision to interpolate nav data to the imagery time base, rather than vice versa, was somewhat arbitrary but makes it easier to use other elements of the images in other analyses.

6.2 Consistent Representation

After temporal alignment, the data are in the form of two sequences of observations:

$\mathbf{N}=\{n(1),n(2),\ldots n(m)\}$ denotes the observations from the nav sensor, and

$\mathbf{E}=\{e(1),e(2),\ldots e(m)\}$denotes the observations from egomotion.

Within a sequence, observations describe in some fashion the trajectory of the sensor and implicitly of the body-fixed coordinate system embedded in the sensor. The trajectory of the other sensor is not identical and is represented in the coordinate system of the other sensor. However, it is known that a paired observation from the two sequences (e.g., $\{n(i)\ e(i)\}, \forall\ i$), is related by the kinematics of being mounted to the same rigid HMMWV body at the same point in time.

In order to compare the sequences from the two sensors in an “apples-to-apples” sense, three steps are necessary. First, we define a notation. Then, using that notation, we describe the

representations of the observations from each sensor. Finally, we relate the manipulations necessary to render the representations consistent for the purpose of comparison.

6.2.1 Notation

Let τ_i^j represent the transformation from coordinate frame *i* to frame *j*, expressed in coordinate frame *i*. Thus, $\tau_{\mathrm{E}}^{\mathrm{N}}$ represents the transformation between the egomotion coordinate frame and the coordinate frame of the nav system, and $\tau_{\mathrm{N}}^{\mathrm{E}}$ its inverse. Since each sensor is affixed to the HMMWV, which is approximately a rigid body, the transformation between the two is time invariant, i.e., constant for any frame *i*, regardless of the instantaneous orientation in world coordinates of the HMMWV.

As we manipulate the observations from the two sensors, we generate two parallel sequences of transformations, one relative to the instantaneous coordinate frame of the nav sensor and one to the instantaneous egomotion coordinate frame. To distinguish between elements of the two sequences, we use a superscript: N for nav, E for egomotion. Therefore, we know, for a paired observation at some index *i* in the sequence,

$$\overset{\mathrm{N}}{\tau}_i^j = \overset{\mathrm{E}}{\tau}_i^j \circ \tau_{\mathrm{E}}^{\mathrm{N}} \text{ , and } \overset{\mathrm{E}}{\tau}_i^j = \overset{\mathrm{N}}{\tau}_i^j \circ \tau_{\mathrm{N}}^{\mathrm{E}},$$

in which

- ∘ is the (non-commutative) coordinate transformation operator so that

$$\tau_i^k = \tau_i^j \circ \tau_j^k \ .$$

We know neither $\tau_{\mathrm{E}}^{\mathrm{N}}$ nor its inverse, however. Furthermore, all the transformations we do know are estimates from noisy sensors.

6.2.2 Representation

The pose data *n(j)* from the nav sensor represent the measured transformation from local coordinates to world coordinates. With the established notation, navigation data elements are of the form $\overset{\mathrm{N}}{\tau}_j^W$. The inverse of a data element *n(j)* is in the more useful form $\overset{\mathrm{N}}{\tau}_W^j$ and is designated *n'(j)*.

The egomotion observation *e(j)* represents the estimated change in pose between image *j* and image (*j+1*), expressed in the coordinate frame of image *j*. This is represented as $\overset{\mathrm{E}}{\tau}_j^{j+1}$. Its inverse *e'(j)* represents the estimated change in pose as one progresses backward in time through the image sequence, and it is of the form $\overset{\mathrm{E}}{\tau}_{j+1}^j$.

6.2.3 Manipulation

A rotation from world coordinates can be expressed as the result of a sequence of frame-to-frame rotations with the notation

$$\tau_W^j = \tau_W^1 \circ \bigwedge_{k=1}^{k=j-1} \tau_k^{k+1}$$

in which

$$\bigwedge_{k=i}^{k=j} \tau_k^{k+1} = \tau_i^{i+1} \circ \tau_{i+1}^{i+2} \circ \cdots \circ \tau_{j-1}^{j}$$

and

τ_W^1 is the initial pose of the sequence.

Egomotion observations of frame-to-frame motion are phenomenologically equivalent to a derivative of the sequence of poses in global coordinates observed by the nav system, so we are taking some liberties with the mathematical terms “differentiate” and “integrate” in order to communicate the approach. To compare the observations, two approaches are possible. First, the frame-to-frame observations from egomotion can be compared to the global transformations observed by nav by the integration of the egomotion observations from the initial pose of the sensor. Since the initial pose is known only in the nav frame, each transformation in the egomotion frame must be transformed to the nav frame.

$$\overset{\mathrm{N}}{\tau}_W^j = \overset{\mathrm{N}}{\tau}_W^1 \circ \bigwedge_{k=1}^{k=j-1} \left(\overset{\mathrm{E}}{\tau}_k^{k+1} \circ \tau_E^N \right) = n'(1) \circ \bigwedge_{k=1}^{k=j-1} \left(e(k) \circ \tau_E^N \right) \tag{3}$$

Without knowledge of the registration transformation, $\tau_{\mathrm{E}}^{\mathrm{N}}$, the transformation from world to local nav coordinates cannot be calculated from egomotion data. However, if $\tau_{\mathrm{E}}^{\mathrm{N}}$ is “near enough” to mathematical identity, the transformation from world coordinates can be approximated over short integral paths, although it will diverge in a cumulative and path-dependent manner.

Alternatively, one can “differentiate” the observations from the nav sensor {*n(j)*}, to compare them with the observations {*e(j)*} from egomotion.

$$\overset{\mathrm{E}}{\tau}_j^{j+1} = \overset{\mathrm{N}}{\tau}_j^W \circ \overset{\mathrm{N}}{\tau}_W^{j+1} \circ \tau_{\mathrm{N}}^{\mathrm{E}} = n(j) \circ n'(j+1) \circ \tau_{\mathrm{N}}^{\mathrm{E}}$$

Without knowledge of the registration, $\tau_{\mathrm{N}}^{\mathrm{E}}$, the comparison is not exact, but the differences are systematic. For a registration that is near enough to identity, the comparison may be meaningful.

Finally, it is important that the integration of egomotion observations is possible only for rotations, since egomotion does not estimate the magnitude of translations—only their direction. The notation developed for full transformations is equally useful for the degenerate case that concerns rotations only.

6.3 Visualization

To facilitate the recognition of consistency between the data streams from the two sensors, paired observations of sensor pose were plotted, corresponding to the time of each image in the sequence. A rotation in three dimensions can be visualized as the magnitude of the rotation and the three-dimensional axis about which the rotation occurs. This rotational axis can be represented as a unit vector. The unit vector is defined in the coordinate frame of the individual sensor, but for the purposes of comparison, we invoke the assumption of near identity rotation between the two coordinate frames, and we defer further discussion of the assumption to section 8. For the purposes of visualizing differences between the two sequences of rotation observations, a quartet of graphs is generated, plotting paired observations of the rotation, indexed by the frame number n, in local coordinates as shown in table 1.

Table 1. Template for figures 2 and 4.

$\theta(n)$, the magnitude of the rotation	$x(n)$, the longitudinal component of the rotation axis unit vector, about which roll occurs
$y(n)$, the transverse component of the rotation axis unit vector, about which pitch occurs	$z(n)$, the vertical component of the rotation axis unit vector, about which yaw occurs

7. Results

First, the incremental rotations are plotted and shown in figure 2. As expected, they are quite noisy, but the rotational axis in particular shows quite clearly that the two sensors are measuring the same phenomenon.

The plots may show evidence of some residual temporal misalignment, but the rotation magnitude error is minimized at this value of temporal offset, so improvement in the alignment is unlikely. The anticipated systematic differences in the rotation axis attributable to rotational misregistration of sensor coordinate frames should be visible in the XY plots of figure 3 as a "skewedness" from the diagonal. However, no such effect is apparent above the noise.

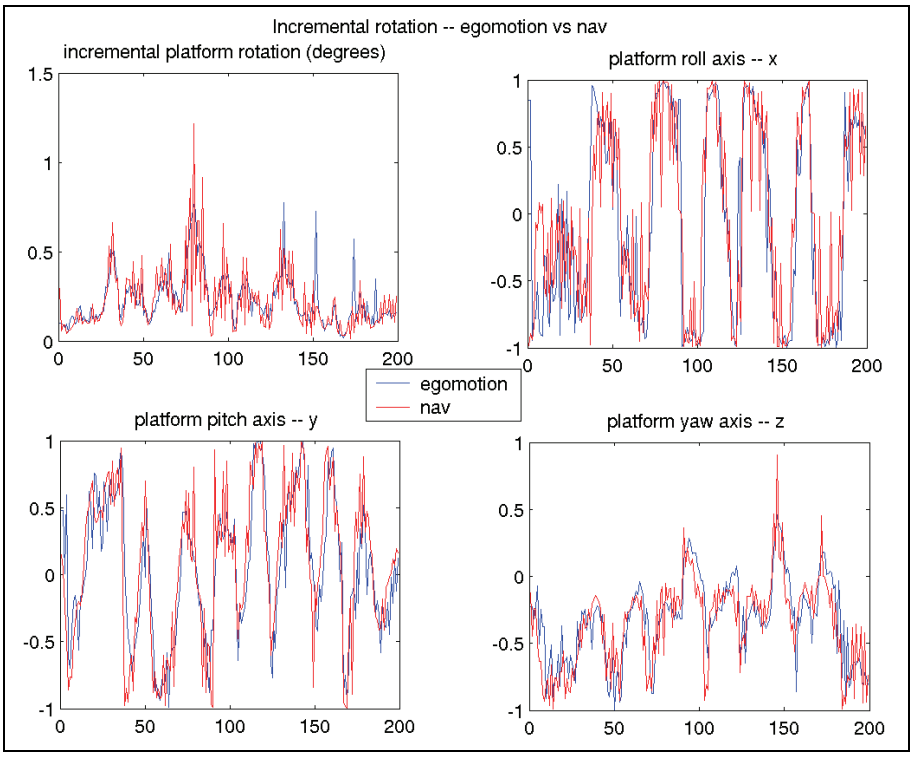

Figure 2. Comparison of incremental rotations. (Roll measured by the nav system exceeded 10 degrees and pitch by 5 degrees during this sequence.)

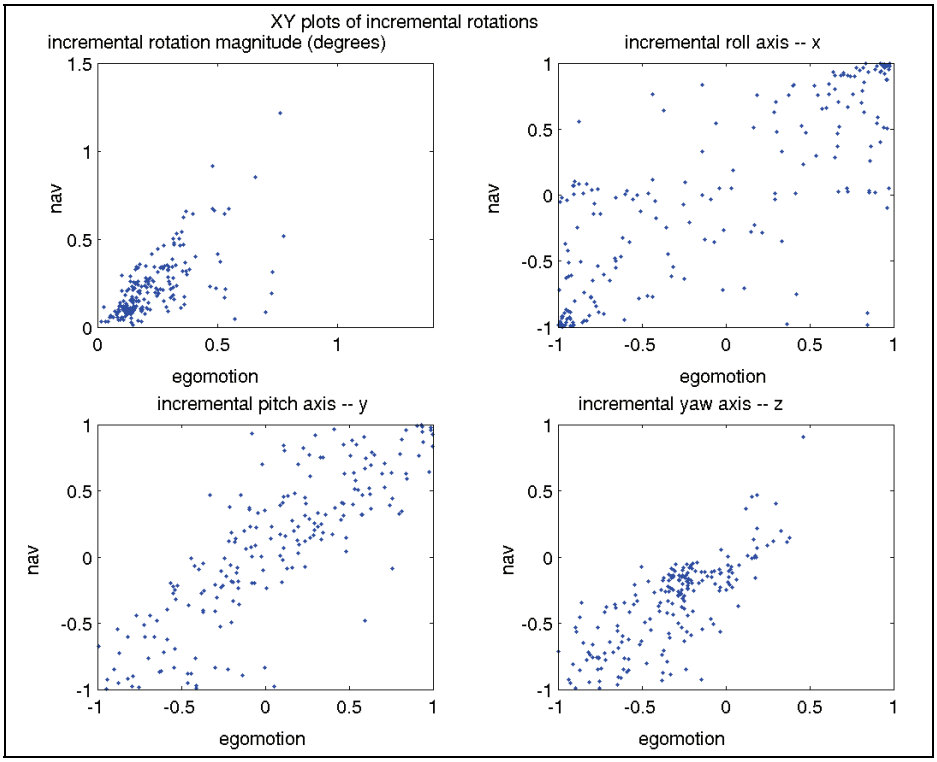

Figure 3. XY plots of incremental rotations.

Now let us compare raw nav rotation observations with egomotion frame-to-frame rotation observations integrated with registration rotation assumed to to be identity, as described in equation 3. Results are shown in figure 4.

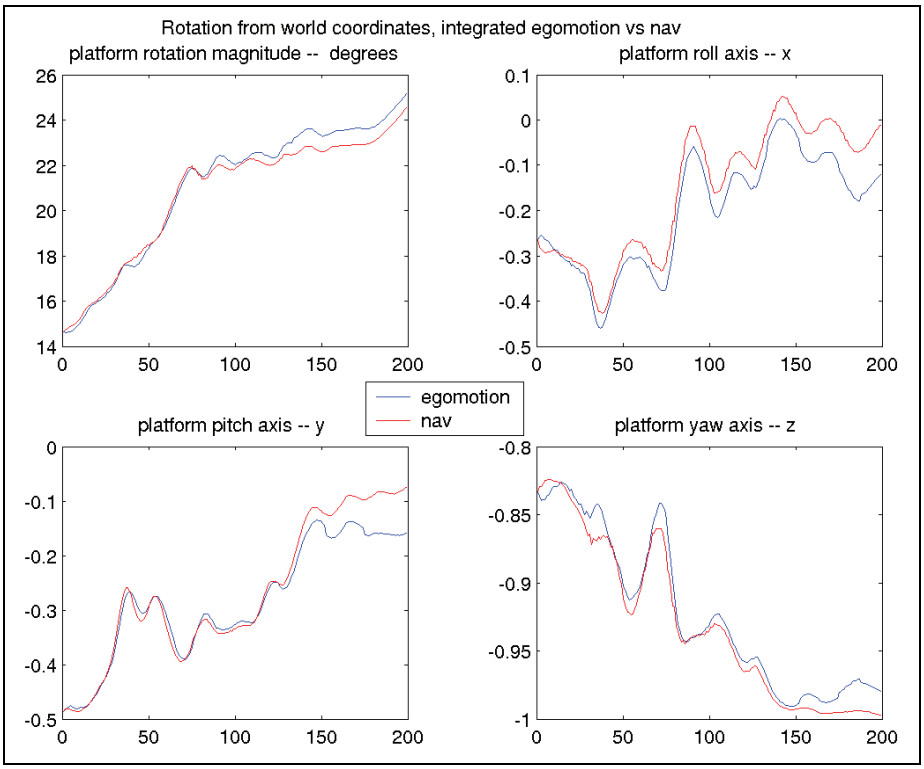

Figure 4. Comparison in world coordinates.

Once again, it is apparent that both sensors are responding to substantially the same phenomenon. As expected, the observations from the two sensors diverge over time, but there is no obvious way to determine how much of that divergence is attributable to violation of the identity registration assumption.

Next, we compare between the two sensors the observations of direction of translation, shown in figure 5. Remember that egomotion does not estimate the magnitude of the translation, only the direction. The direction as expected is primarily in the longitudinal ("forward") direction of the HMMWV carrying the sensors, but beyond that, the egomotion observations do not track the nav measurements particularly well. As expected, the transverse translation is small, constrained by the steering effect of the wheels and tires. The nav system estimates the transverse translation as nearly zero, but it is possible that the dead reckoning-based nav observations do not account for transverse motion resulting from the yaw acting along the vehicle wheelbase. It is also possible that the transverse excursions recorded by the egomotion measurements are rotation effects attributable to the position of the camera at the opposite end of the vehicle from the rear axle where odometry is recorded.

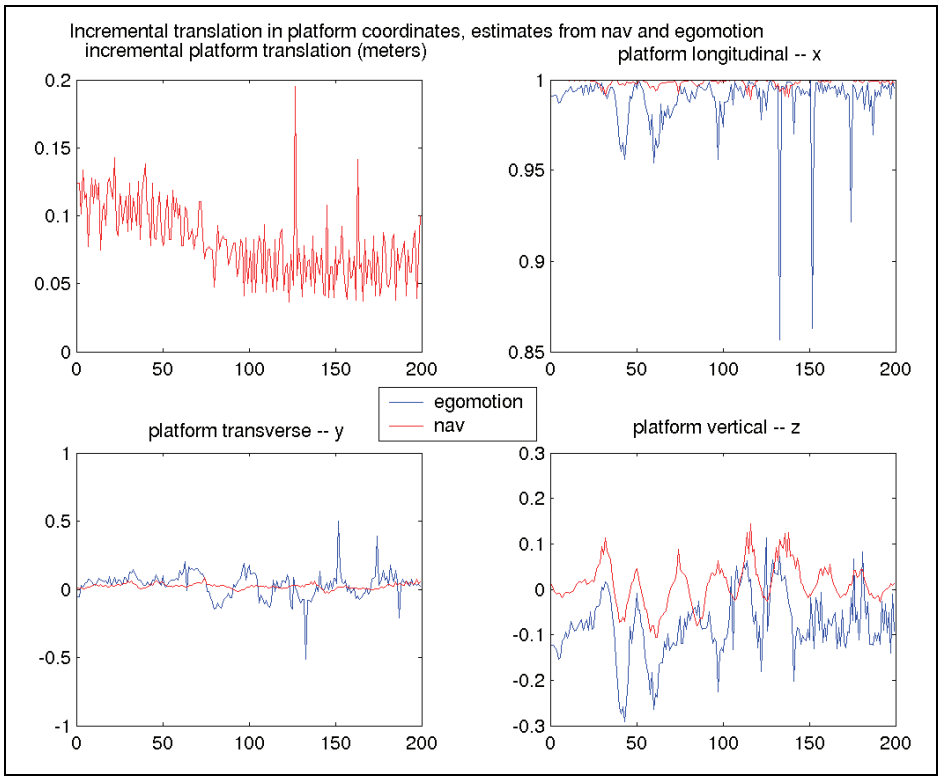

Figure 5. Comparison of incremental translation.

A qualitatively significant effect is apparent in the vertical translations measured by the two sensors. The nav system senses that its trajectory is roughly horizontal in its coordinate frame, while egomotion senses that it is consistently climbing (a negative translation in the "down" direction). This is consistent with a camera axis tilted below the vehicle horizontal (the so-called "look-down angle" used to concentrate camera field of view on the ground rather than uselessly on the sky) during forward vehicle motion. The camera system is known to have been so tilted, so the results are as expected.

8. Registration From Trajectory

The difference in the coordinate frames of the two sensors can, in principle, be seen in the difference in representation of an incremental transformation as seen by the two sensors. In other words, $\tau_{\mathrm{E}}^{\mathrm{N}}$ should be evident in the juxtaposition of $\overset{\mathrm{E}}{\tau}_{j}^{j+1}$ and $\overset{\mathrm{N}}{\tau}_{j}^{j+1}$, for any frame j. We have seen evidence in the "vertical" element of juxtaposed incremental translations, but translations of different points of a rotating rigid body are not necessarily the same. A kinematically cleaner place to observe it is in the rotation axis of incremental rotations because the rotation of any point on a rigid body is identical.

So let us from this point on consider only the rotational element of our transformations, τ_i^j. For an incremental rotation, $\overset{\mathrm{N}}{\tau}{}_j^{j+1}$, we know that $\overset{\mathrm{N}}{\tau}{}_j^{j+1} = \overset{\mathrm{E}}{\tau}{}_j^{j+1} \circ \tau_{\mathrm{E}}^{\mathrm{N}}$, so for each pair of observations, we can calculate directly an estimate of the rotational registration between sensors:

$$\begin{aligned} \tau_{\mathrm{E}}^{\mathrm{N}} &= \overset{\mathrm{E}}{\tau}{}_{j+1}^{j} \circ \overset{\mathrm{N}}{\tau}{}_j^{j+1} = e'(j) \circ \left(\overset{\mathrm{N}}{\tau}{}_j^{W} \circ \overset{\mathrm{N}}{\tau}{}_W^{j+1} \right). \\ &= e'(j) \circ (n(j) \circ n'(j+1)) \end{aligned} \tag{4}$$

However, implementation of this computation on data from the HS yielded estimates of registration which were not credible, varying substantially frame to frame rather than describing random variation about a single rotation as expected.

8.1 Registration From Trajectory With Synthetic Data

A small experiment was conducted with synthetic data to examine the effectiveness of computing registration from trajectory. Data from a synthetic sensor were created by rotation of the axis of the rotation data from the HS egomotion set by a known rotation (1 degree about the pitch axis). The data were equivalent to a "perfect" (all noise common mode) second egomotion sensor on the same rigid platform, with known registration. The registration was then back calculated with equation 4 and plotted in figure 6. The error in the computed registration was consistently greater when the axis of motion of the platform coincided with the axis of rotation of the registration, e.g., when motion of the platform was substantially a pitch motion. The data were then edited to eliminate instances of nearly pure pitch motions by deletion of data points where the pitch component of the unit rotation axis was outside the range (-0.5 0.5). The remaining registrations were much more correct, as shown in figure 7. If we can correctly "guess" the registration angle of real sensors and if we can cull motions near that angle, we may be able to confirm our guess. Evaluating the effectiveness of this approach is beyond the scope of this study, but a necessary condition for its success appears satisfied.

The registration is calculated for each of the 200 pairs of data. The top plot of the triple shows the magnitude of the calculated registration; the middle plot shows the components of the unit rotation axis. The third plot of the triple shows the pitch component of the egomotion.

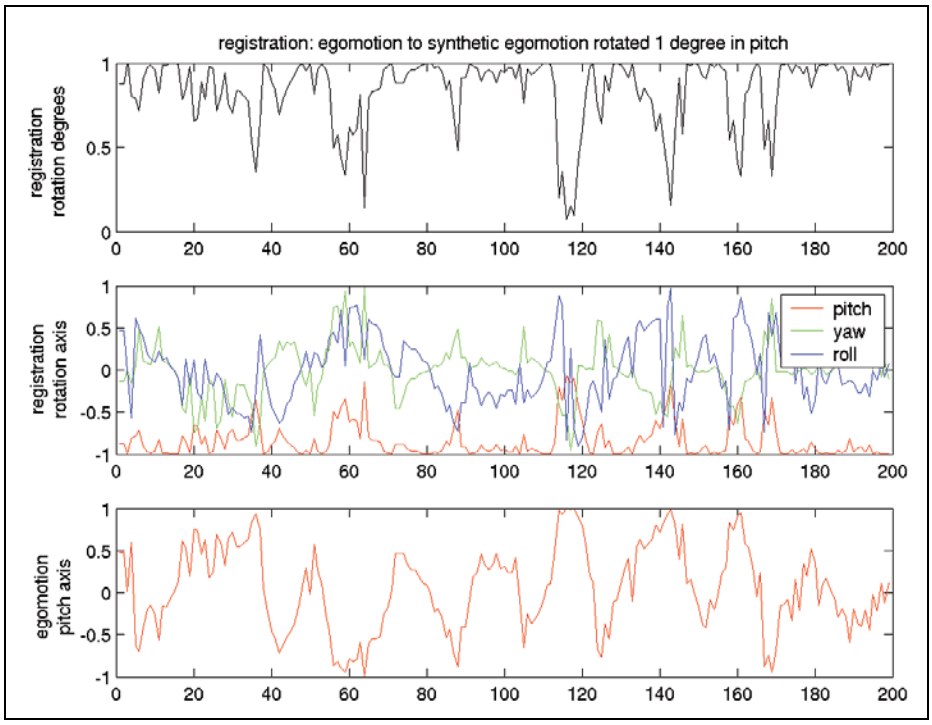

Figure 6. Registration of synthetic sensor from trajectory.

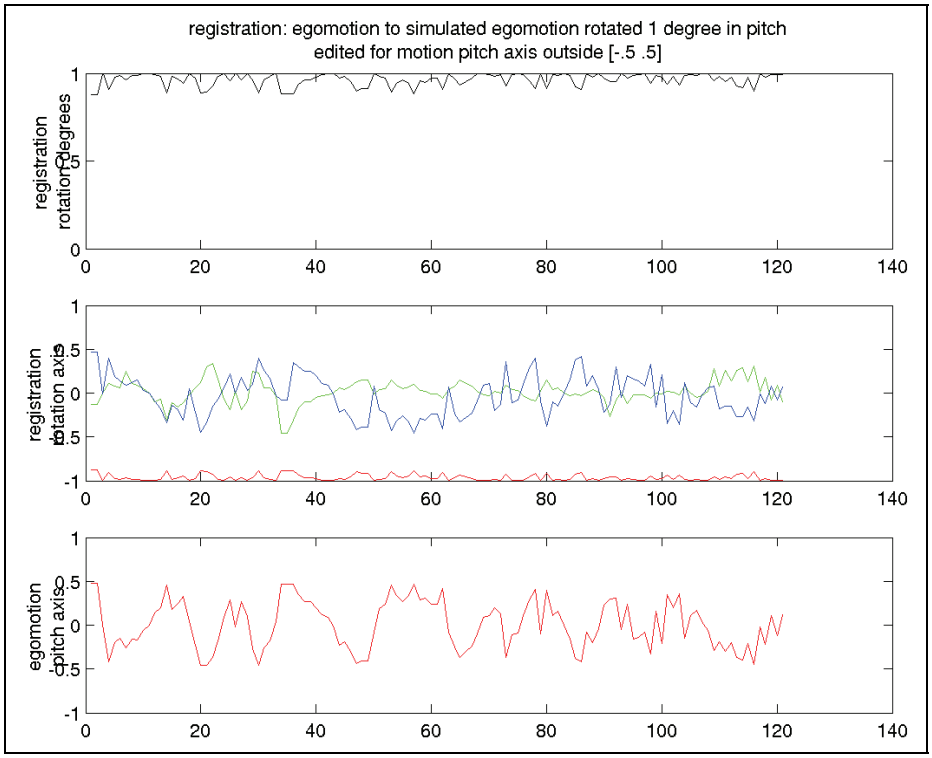

Figure 7. Registration of synthetic sensor from the same trajectory purged of motions near the known registration axis. (The magnitude of the computed registration rotation is approximately 1 degree, as expected, and the direction is almost pure pitch, again as expected.)

9. Conclusions

1. In the data set considered in this study, the rotational egomotion calculated from camera imagery approximates navigation measurements taken concurrently. Both measurements are noisy, however, and further work fusing the two should include signal processing and/or statistical methods to smooth the signals and/or eliminate outliers. Another source of noise is temporal misalignment of the two recorded data streams. Kinematics say the magnitude of the rotation angle sensed by the two sensors ought to be invariant to the registration. Selecting the offset in time base, which minimizes the differences between the estimates of this parameter, is one way to align the data streams, and this data set delivered a credible although unverifiable result.

2. Registration from trajectory appears to be much more difficult than the formulation of the problem would imply. It is worth pursuing, however, because no other way is apparent to register the imaging sensors (with which the UGV plans its path across terrain) with the navigation system that tracks the trajectory that the UGV follows. Simple calculation of the registration does not look promising, so other approaches (such as the one presented in section 8.1) should be investigated.

3. In the absence of independent ground truth, there is nothing upon which to base a judgment concerning translation measurements except the agreement between the sensors, and the translational trajectories do not track well. Some dissimilarities are to be expected because of kinematics, since the two sensors were not co-located. We are not convinced of the ability of dead reckoning to adequately account for small translations in the transverse and vertical directions. Future work in fusion should, if possible, use a nav sensor with explicit measurement of transverse and vertical motion components.

4. In general, agreement between noisy sensors is an unsatisfactory measure of success. It is unclear how ground truth for this study could have been measured, so accuracy of the measurement may not be easy to determine. It should be possible to define a measure that can be compared for consistency across the iterations, however. For example, three-dimensional reconstructions of point features can be tracked across imagery sequences to estimate the precision, if not the accuracy, of egomotion estimates. Future studies should endeavor to identify and implement such measures.

10. References

Harris, C.G.; Stephens, M. A Combined Corner and Edge Detector, Proceedings of the 4th Alvey Vision Conference, 1988, 147–151, Manchester, United Kingdom.

Honeywell, MAPS datasheet, http://www.ais.honeywell.com/dss/sgp/datasheets/gn-maps_datasht.pdf, no date.

Oberle, W.F. Stereo Camera Re-calibration and the Impact of Pixel Location Uncertainty, Army Research Laboratory Technical Report; ARL-TR-2979; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, May 2003.

Oberle, W.F.; Haas, G.A. Three-Dimensional Stereo Reconstruction and Sensor Registration With Application to the Development of a Multi-Sensor Database; ARL-TR-2878; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, December 2002.

Shneier M.; Chang, T.; Hong, TH. Repository of sensor data for autonomous driving research, Unmanned Ground Vehicle Technology V, *Proceedings of SPIE,* **April 2003**, *5083*, 22–23.

Shoemake, K. Animating Rotation with Quaternion Curves, *Proceedings of SIGGRAPH* 85, **July 22-26 1985**, *19*(3), 245–254.

Shoemaker, C.M.; Bornstein, J.A. The Demo III UGV program: a testbed for autonomous navigation research, *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC)*, **14-17 Sept. 1998**, 644–651.

Torr, P.H.S. *A Structure and Motion Toolkit in MATLAB, Interactive Adventures in S and M;* Technical Report MSR-TR-2002-56; Microsoft Research: Cambridge, United Kingdom, June 2002 (web site: http://research.microsoft.com/~philtorr/).

Trucco, E.; Verri, A. *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.

Wang, W.; Tsui, H.T. A SVD Decomposition of Essential Matrix with Eight Solutions for the Relative Positions of Two Perspective Cameras, *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, **September 2000**; *1*, 362–365.

INTENTIONALLY LEFT BLANK

Appendix A. MATLAB[7] Code to Plot Juxtaposed Translations

```
% NavEgoTran.m
% This routine compares the translations sensed by the nav system
% with translations detected by calculating the egomotion of the left camera
% of a stereo pair.  The sensor data came from a sequence corresponding to
% frames 30300 to 30499 (approx 7 seconds) of stereo imagery.

% this version compares incremental translations between nav and egomotion
% approximately in the platform coordinate frame

% First get the nav data, contained in "navVector"
% The variables start1, end1, offset, navfname, and vsfname confirm the source of the navVector
data
% and are not used in this routine
load 'nav.mat';
rot2=[navVector.rot];
rot4=reshape(rot2,4,200)';      % rot4 is rotation from body-fixed to world
flipr=-1*eye(4);flipr(1,1)=1;
rot3=rot4*flipr;                % rot3 is from world to body-fixed -- the direction the truck is facing

% now calculate the frame-to-frame translations reported by nav
tran=[navVector.tranRel];
nFrames=length(tran)/3;
tran2=reshape(tran,3,nFrames)';
deltaTran(1:nFrames-1,:)=tran2(2:nFrames,:)-tran2(1:nFrames-1,:);
for i1=1:nFrames-1,dist(i1)=norm(deltaTran(i1,:)); end

T4=zeros(length(deltaTran),3);

% rotate the translation vector to the coordinate frame of the sensor
% (so the vector coincides roughly with platform coordinates)
for i=1:length(deltaTran)
   R1=rot3(i,:);
   R2=q2R(R1);
   p1=deltaTran(i,:)';
   p2=(qpMult(R1,p1));
   T3(i)=norm(p2);
   T4(i,:)=p2/norm(p2);
end

% Now fetch the egomotion data, in Rot (Rotation matrices for each frame, not used here),
%  T (translation vector for each frame), and q (quaternion form of rotations)
egoMatFile='egoL.mat'
```

[7] MATLAB® is a registered trademark of The MathWorks.

```
clear egoMatFileNew;
egoMatFileNew=input('input mat file containing egomotion data, in single quotes')
if length(egoMatFileNew)>0 egoMatFile=egoMatFileNew,end
load(egoMatFile);
qp=zeros(size(q));
qpInNavCoorsd=zeros(size(qp));

figure,subplot(2,2,1),plot(T3(:),'red'),title('incremental platform translation (meters) ')
% interchange axes to compare similar
subplot(2,2,3),plot(T(:,1)),hold,plot(T4(:,2),'red'),title('platform transverse -- y ')
subplot(2,2,4),plot(T(:,2)),hold,plot(T4(:,3),'red'),title('platform verticall -- z ')
subplot(2,2,2),plot(T(:,3)),hold,plot(T4(:,1),'red'),title('platform longitudinal -- x ')
```

Appendix B. MATLAB Code to Plot Juxtaposed Incremental Rotations

```
% navEgoInc.m
% This routine compares the rotations sensed by the nav system
% with rotations detected by calculating the egomotion of the left camera
% of a stereo pair.  The sensor data came from a sequence corresponding to
% frames 30300 to 30499 (approx 7 seconds) of stereo imagery.

% this differs from navEgo in that the rotations considered are incremental,
% not integrated over the sequence

% First get the nav data, contained in “navVector”
% The variables start1, end1, offset, navfname, and vsfname confirm the source of the navVector
data
% and are not used in this routine

load('nav.mat')
rot2=[navVector.rot];
rot4=reshape(rot2,4,200)';      % rot4 is rotation from body-fixed axis back to world coords
flipr=-1*eye(4);flipr(1,1)=1;
rot3=rot4*flipr;                % rot3 is rotation from world coords to body-fixed

% now differentiate, defining rotations from frame j to (j+1) in rot5
rot5=zeros(size(rot4));
for i=1:length(rot5)-1
   qq2=rot4(i,:);qq1=rot3(i+1,:);  % first comes a rotation from frame j to world, then from world
to frame j+1
   rot5(i,:)=qqMult(qq1,qq2);      % qqMult(q1,q2) uses quaternion multiplication notation,
means q2 happens first
end
rot5=rot5(1:length(rot5)-1,:);      % no increment for the last frame

% Now fetch the egomotion data, in Rot (Rotation matrices for each frame, not used here),
%  T (translation vector for each frame), and q (quaternion form of rotations)
egoMatFile='egoL.mat'
clear egoMatFileNew;
egoMatFileNew=input('input mat file containing egomotion data, in single quotes')
if length(egoMatFileNew)>0 egoMatFile=egoMatFileNew,end
load(egoMatFile);

% isolate and normalize the quaternian rotation vector for plotting
for i=1:(length(q))
   i1=i;
   rotV(i1,:)=rot5(i1,2:4)/norm(rot5(i1,2:4));
   qpV(i1,:)=q(i1,2:4)/norm(q(i1,2:4));
end
```

```
% convert w to degrees and plot juxtaposed incremental rotations
rot6=2*acos(rot5(:,1))/pi*180;q6=2*acos(q(:,1))/pi*180;
figure,subplot(2,2,1),plot(q6(:)),hold,plot(rot6(:),'red'),title('incremental platform rotation
(degrees) ')
subplot(2,2,2),plot(qpV(:,1)),hold,plot(rotV(:,2),'red'),title('platform pitch axis -- y ')
subplot(2,2,3),plot(qpV(:,2)),hold,plot(rotV(:,3),'red'),title('platform yaw axis -- z ')
subplot(2,2,4),plot(qpV(:,3)),hold,plot(rotV(:,1),'red'),title('platform roll axis -- x ')

% now in XY plot form
plot1=q6;plot2=rot6;
figure,subplot(2,2,1),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental
rotation magnitude (degrees)')
plot1=qpV(:,3);plot2=rotV(:,1);
subplot(2,2,2),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental roll axis -- x
')
plot1=qpV(:,1);plot2=rotV(:,2);
subplot(2,2,3),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental pitch axis --
y ')
plot1=qpV(:,2);plot2=rotV(:,3);
subplot(2,2,4),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental yaw axis -- z
')

% calculate and plot computed registration
for i=1:length(qpV)
   v1(1)=qpV(i,3);v1(2)=qpV(i,1);v1(3)=qpV(i,2);
   v2=rotV(i,:);
   s=acos(dot(v1,v2));
   vx=cross(v1,v2);
   sC(i)=s/pi*180;;
   vC(i,:)=vx/norm(vx);
end

figure,subplot(2,1,1);plot(sC(:),'black');title('registration'),subplot(2,1,2);plot(vC(:,1),'red');hold;p
lot(vC(:,2),'green');
plot(vC(:,3),'blue');hold off
```

Appendix C. MATLAB Code to Plot Juxtaposed Absolute Rotations

```
% navEgoAbs.m
%  This routine compares the rotations sensed by the nav system
% with rotations detected by calculating the egomotion of the left camera
% of a stereo pair.  The sensor data came from a sequence corresponding to
% frames 30300 to 30499 (approx 7 seconds) of stereo imagery.

% this differs from navEgo in that the rotations considered are incremental,
% not integrated over the sequence

% First get the nav data, contained in “navVector”
% The variables start1, end1, offset, navfname, and vsfname confirm the source of the navVector
data
% and are not used in this routine

load('nav.mat')
rot2=[navVector.rot];
rot4=reshape(rot2,4,200)';      % rot4 is rotation from body-fixed axis back to world coords
flipr=-1*eye(4);flipr(1,1)=1;
rot3=rot4*flipr;                % rot3 is rotation from world coords to body-fixed

% now differentiate, defining rotations from frame j to (j+1) in rot5
rot5=zeros(size(rot4));
for i=1:length(rot5)-1
   qq2=rot4(i,:);qq1=rot3(i+1,:);  % first comes a rotation from frame j to world, then from world
to frame j+1
   rot5(i,:)=qqMult(qq1,qq2);      % qqMult(q1,q2) uses quaternion multiplication notation,
means q2 happens first
end
rot5=rot5(1:length(rot5)-1,:);      % no increment for the last frame

% Now fetch the egomotion data, in Rot (Rotation matrices for each frame, not used here),
%  T (translation vector for each frame), and q (quaternion form of rotations)
egoMatFile='egoL.mat'
clear egoMatFileNew;
egoMatFileNew=input('input mat file containing egomotion data, in single quotes')
if length(egoMatFileNew)>0 egoMatFile=egoMatFileNew,end
load(egoMatFile);

% isolate and normalize the quaternian rotation vector for plotting
for i=1:(length(q))
   i1=i;
   rotV(i1,:)=rot5(i1,2:4)/norm(rot5(i1,2:4));
   qpV(i1,:)=q(i1,2:4)/norm(q(i1,2:4));
end
```

```
% convert w to degrees and plot juxtaposed incremental rotations
rot6=2*acos(rot5(:,1))/pi*180;q6=2*acos(q(:,1))/pi*180;
figure,subplot(2,2,1),plot(q6(:)),hold,plot(rot6(:),'red'),title('incremental platform rotation
(degrees) ')
subplot(2,2,2),plot(qpV(:,1)),hold,plot(rotV(:,2),'red'),title('platform pitch axis -- y ')
subplot(2,2,3),plot(qpV(:,2)),hold,plot(rotV(:,3),'red'),title('platform yaw axis -- z ')
subplot(2,2,4),plot(qpV(:,3)),hold,plot(rotV(:,1),'red'),title('platform roll axis -- x ')

% now in XY plot form
plot1=q6;plot2=rot6;
figure,subplot(2,2,1),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental
rotation magnitude (degrees)')
plot1=qpV(:,3);plot2=rotV(:,1);
subplot(2,2,2),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental roll axis -- x
')
plot1=qpV(:,1);plot2=rotV(:,2);
subplot(2,2,3),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental pitch axis --
y ')
plot1=qpV(:,2);plot2=rotV(:,3);
subplot(2,2,4),plot(plot1,plot2,'.'),xlabel('egomotion'),ylabel('nav'),title('incremental yaw axis -- z
')

% calculate and plot computed registration
for i=1:length(qpV)
   v1(1)=qpV(i,3);v1(2)=qpV(i,1);v1(3)=qpV(i,2);
   v2=rotV(i,:);
   s=acos(dot(v1,v2));
   vx=cross(v1,v2);
   sC(i)=s/pi*180;;
   vC(i,:)=vx/norm(vx);
end

figure,subplot(2,1,1);plot(sC(:),'black');title('registration'),subplot(2,1,2);plot(vC(:,1),'red');hold;p
lot(vC(:,2),'green');
plot(vC(:,3),'blue');hold off
```

NO. OF
COPIES ORGANIZATION

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL CI IS R REC MGMT
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL CI OK TECH LIB
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL D D SMITH
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL SE SE N NASRABADI
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 NATL INST OF STDS & TECHNOLOGY
ATTN DR M SHNEIER
BLDG 200
GAITHERSBURG MD 20899

1 SARNOFF CORP
ATTN M/S W-357 DR R MANDELBAUM
201 WASHINGTON RD
PRINCETON NJ 08543-5300

1 CARNEGIE MELLON UNIV
ROBOTICS INST DR M HEBERT
5000 FORBES AVE
PITTSBURGH PA 15213

1 SRI INTERNATL
ATTN DR K KONOLIGE
333 RAVENSWOOD AVE
MENLO PARK CA 94025-3493

1 UNIV OF MARYLAND
INST FOR ADV COMPUTER STUDIES
ATTN DR L DAVIS
COLLEGE PARK MD 20742-3251

1 GENERAL DYNAMICS ROBOTIC SYS
ATTN J KURTZ
1234 TECH COURT
WESTMINSTER MD 21167-3029

NO. OF
COPIES ORGANIZATION

1 PERCEPTEK LLC
ATTN M ROSENBLUM
9892 TITAN CT UNIT 7
LITTLETON CO 80125

1 JET PROPULSION LABORATORY
MACHINE VISION GROUP
ATTN DR L MATTHIES
4800 OAK GROVE DR
PASADENA CA 91109

ABERDEEN PROVING GROUND

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL CI OK (TECH LIB)
BLDG 305 APG AA

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL HR SC C STACHOWIAK
BLDG 459

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL WM J SMITH
T ROSENBERGER
BLDG 4600

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL WM B A HORST
W CIEPIELLA
BLDG 4600

3 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL WM BA D LYON
A THOMPSON T VONG
BLDG 4600

15 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL WM BF J LACETERA
H EDGE P FAZIO
M FIELDS G HAAS (4)
T HAUG W OBERLE (4)
R PEARSON R VON WAHLDE
BLDG 390

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRL WM RP C SHOEMAKER
J BORNSTEIN
BLDG 1121